CLAIMS

Pub B3

1. A method for decoding a first composite packet in a processor, said method comprising the steps of:

providing assembly code for each one of a plurality of instructions in a first combination of instructions in said first composite packet;

matching a template in said first composite packet to a known template corresponding to one of a plurality of known syntaxes;

matching said one of said plurality of known syntaxes with a resolved packet syntax;

using said resolved packet syntax to determine assembly code associated with execution of said first combination of instructions;

providing assembly code associated with execution of said first combination of instructions.

2. The method of claim 1 wherein said step of matching said one of said plurality of known syntaxes comprises the step of matching each term in said one of said plurality of known syntaxes against a respective term in said resolved packet syntax.

3. The method of claim 2 wherein said matching step is a direct matching step.

4. The method of claim 1 wherein said assembly code associated with execution of said first combination of instructions specifies an issue group for said first

combination of instructions.

5. The method of claim 1 wherein said assembly code associated with execution of said first combination of instructions specifies a plurality of issue groups for said first combination of instructions.

6. The method of claim 1 wherein said assembly code associated with execution of said first combination of instructions identifies a chained instruction in said first combination of instructions, wherein said chained instruction belongs to an issue group in a second composite packet.

7. The method of claim 1 wherein said assembly code associated with execution of said first combination of instructions identifies a plurality of chained instructions in said first combination of instructions, wherein said plurality of chained instructions belong to respective issue groups in a second composite packet.

8. The method of claim 1 wherein said plurality of known syntaxes are arranged as a plurality of first level nodes in a tree structure.

9. The method of claim 1 wherein said known template identifies at least one issue group in said first composite packet.

10. The method of claim 1 wherein said known template identifies a chained instruction in said first combination of instructions, wherein said chained instruction belongs to an issue group in a second composite packet.

11. The method of claim 1 wherein said known template identifies a plurality of chained instructions in said first combination of instructions, wherein said plurality of chained instructions belong to respective issue groups in a second composite packet.

12. The method of claim 3 wherein said plurality of known syntaxes are arranged as a plurality of first level nodes in a tree structure and said direct matching step comprises matching a term in said resolved packet syntax with a term in a syntax of one of said plurality of first level nodes.

13. The method of claim 1 wherein said composite packet in said processor consists of 128 bits.

14. The method of claim 1 wherein said composite packet in said processor consists of 256 bits.

15. The method of claim 1 wherein each instruction in said first combination of instructions consists of 16 bits.

16. The method of claim 1 wherein each instruction in said first combination of instructions consists of 32 bits.

17. The method of claim 1 wherein each instruction in said first combination of instructions consists of 41 bits.

18. The method of claim 1 wherein said first combination of instructions comprises at least two instructions.

19. The method of claim 1 wherein said first combination of instructions comprises at least one issue group.

20. The method of claim 19 wherein said at least one issue group comprises at least one instruction.

21. The method of claim 1 wherein said template comprises at least five bits.

22. A method for simulating execution of a composite packet in a processor, said method comprising the steps of:

simulating a fetching of said composite packet;

decoding said composite packet so as to determine assembly code associated with execution of a first combination of instructions in said composite packet;

issuing an individual instruction from said first combination of instructions according to said assembly code associated with execution of said first combination of instructions;

simulating allocation of an execution unit to said individual instruction;

simulating execution of said individual instruction in said execution unit.

23. The method of claim 22 wherein said step of simulating said fetching comprises:

comparing a number of composite packets in a packet queue with a queue length of said packet queue;

placing said composite packet in said packet queue when said number of composite packets in said packet queue is less than said queue length;

delaying said step of simulating said fetching when said number of composite packets in said packet queue is equal to said queue length.

24. The method of claim 23 wherein said queue length is 2.

25. The method of claim 23 further comprising steps of:

determining whether each composite packet in said packet queue is available based on a fetch latency;

delaying said decoding step until said composite packet has been determined to be available.

26. The method of claim 25 wherein said fetch latency is 1 cycle.

27. The method of claim 22 wherein said decoding step comprises:

providing assembly code for each one of a plurality of instructions in a first combination of instructions in said first composite packet;

matching a template in said first composite packet to a known template corresponding to one of a plurality of known syntaxes;

matching said one of said plurality of known syntaxes with a resolved packet syntax;

using said resolved packet syntax to determine assembly code associated with execution of said first combination of instructions;

providing assembly code associated with execution of said first combination of instructions.

28. The method of claim 22 wherein said issuing step comprises:

comparing a number of individual instructions in an instruction window with an instruction window size of said instruction window;

placing said individual instruction from said first combination of instructions in said instruction window when said number of individual instructions in said instruction window is less than said instruction window size;

delaying said issuing step when said number of individual instructions in said instruction window is equal to said instruction window size.

29. The method of claim 22 wherein said assembly code associated with execution of said first combination of instructions specifies an issue group for said first combination of instructions.

30. The method of claim 29 wherein said issuing step comprises:

comparing a number of individual instructions in an instruction window with an instruction window size of said instruction window;

determining a number of spaces in said instruction window by subtracting said number of individual instructions in said instruction window from said instruction window size;

determining a number equal to the number of individual instructions in said issue group;

placing said individual instruction from said first combination of instructions in said instruction window when said number of individual instructions in said issue group is less than or equal to said number of spaces in said instruction window;

delaying said issuing step when said number of spaces in said instruction window is less than said number of individual instructions in said issue group.

31. The method of claim 22 wherein said assembly code associated with execution of said first combination of instructions identifies a chained instruction from said first combination of instructions, wherein said chained instruction belongs to an issue group in a second composite packet.